

Predictive Models

CS-370: Software Design and Development

Robert Bruce

Predictive development model versus Adaptive development model

- **Predictive development model:** anticipate the project needs based on requirements then proceed to complete the project.
- **Adaptive development model:** This model allows for malleable project goals which can be changed during the development phase.

Predictive development models: Advantages

- **Predictability:** Everything is planned out. You know when the project will be finished.
- **Stability:** Requirements gathering committed at beginning of project. Customer knows what they will receive when project is finished.
- **Cost-savings:** Developers won't waste time on irrelevant project details which lead to dead-ends.
- **Detailed-design:** No decision making necessary in later stages of development. The decision making occurs up-front in planning stage.
- **Less refactoring** (compared to adaptive models): Code refactoring (or code rewriting) is minimized in predictive models.

Predictive development models: Advantages (continued)

- **Fix bugs early:** With detailed requirements and design completed, software bugs can be discovered early in the development process.
- **Better documentation** (compared to adaptive models): predictive models demand detailed documentation before the development process begins. Adaptive models de-emphasize documentation and design commitment as a trade-off to support adaptability and flexibility in development.
- **More time to prepare training materials:** The time spent on documentation before development of the project gives the project team time during the development phase to write training materials since the documentation is already committed before development begins.
- **Easy maintenance:** A well-planned, detailed project is consequently easy to maintain since the solution was developed elegantly and from a wide-perspective.

Predictive development models: Disadvantages

- **Inflexible:** Making requirements changes is difficult once the project is in development.
- **Later initial release** (compared to adaptive models): Customers don't see a prototype but rather must wait until the final product is finished.
- **Big Design Up Front (BDUF):** creates a dependency back-log where developers are waiting for other developers to complete tasks before other tasks can be started.

Predictive development models: Qualities that lead to success

The following qualities will likely lead to success in predictive models:

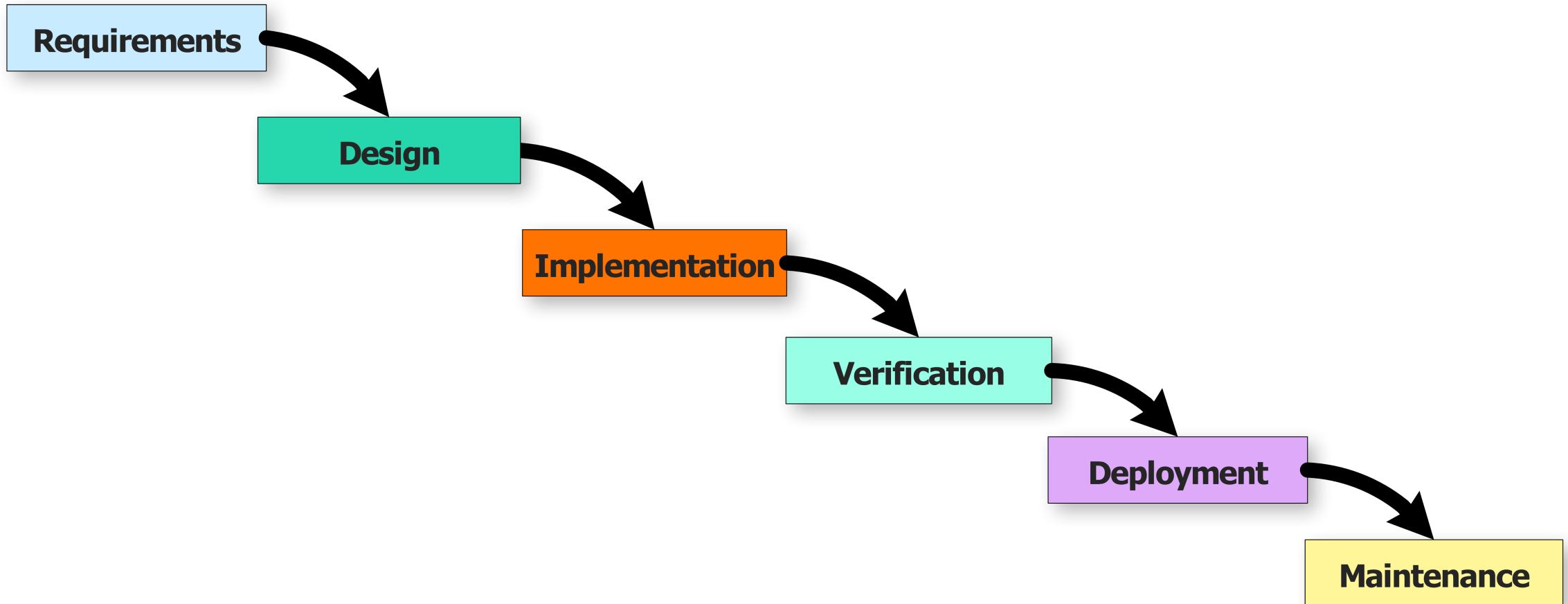
- **User involvement:** Users contribute to the project requirements.
- **Clear vision:** Customer and developer visions are aligned in the project goals.
- **Limited size:** The project is small enough that everyone can see the big picture. Consequently, project requirements do not change over time.
- **Experienced team:** Team developers have sufficient knowledge to take the project to completion.
- **Realistic:** The project objectives are reasonable, supportable, and tenable.
- **Established technology:** The team developing the project utilizes technology they have prior experience with.

Predictive development models: Qualities that lead to failure

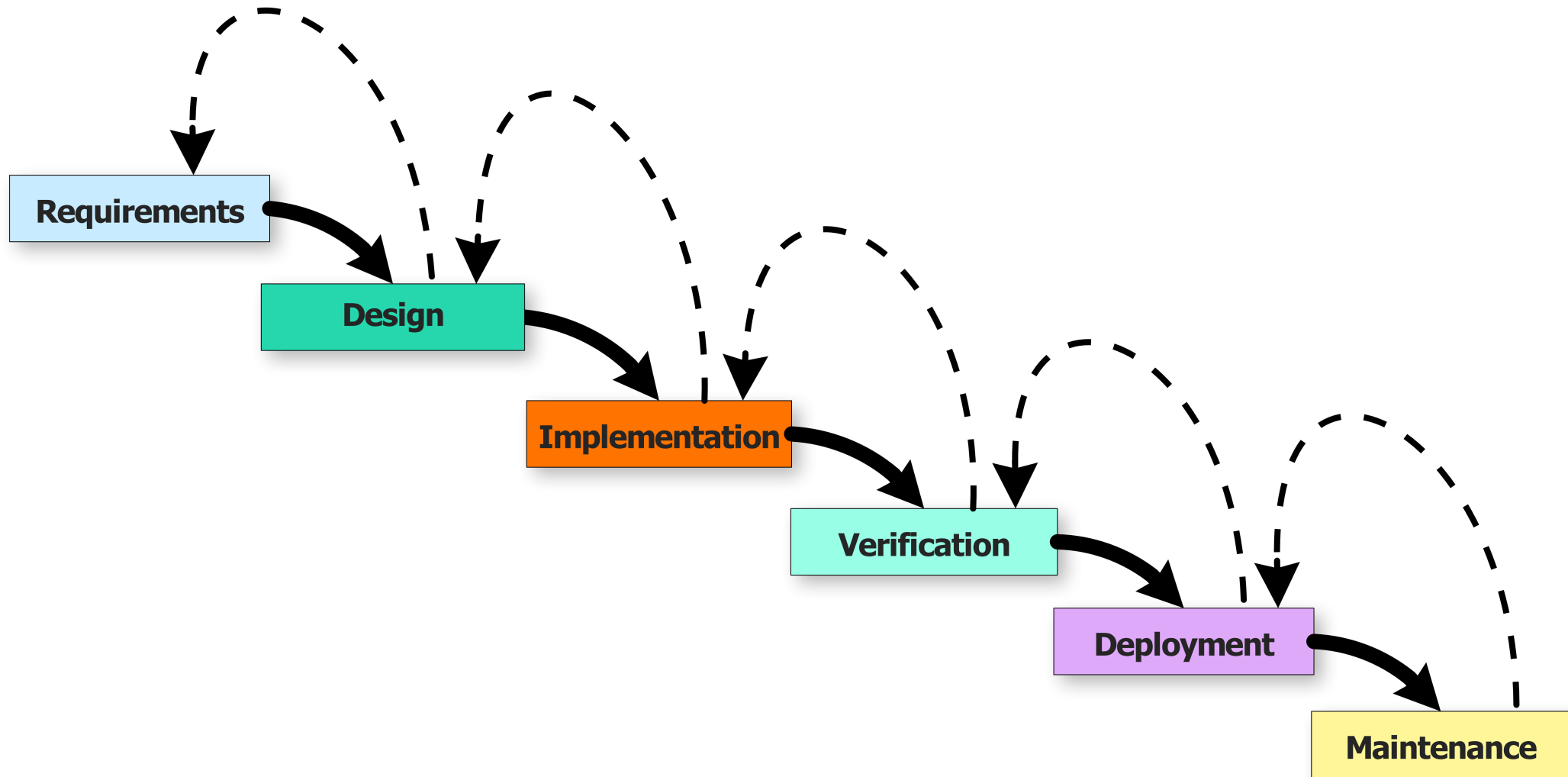
The following qualities will likely lead to failure in predictive models:

- **Incomplete requirements:** A project cannot be completed in a predictive model if details about each requirements are vague or missing.
- **Unclear requirements:** There is little or no consensus between customers and developers concerning the project requirements. Consequently, the customer will not be satisfied with the resultant project.
- **Changing requirements:** Evolving requirements changes the predictability and the inherent developmental priorities of a project. Consequently, the project will likely go over-budget (money), be delayed (time), or both.
- **Insufficient resources:** An experienced developer is still limited in how much they can achieve within one week.

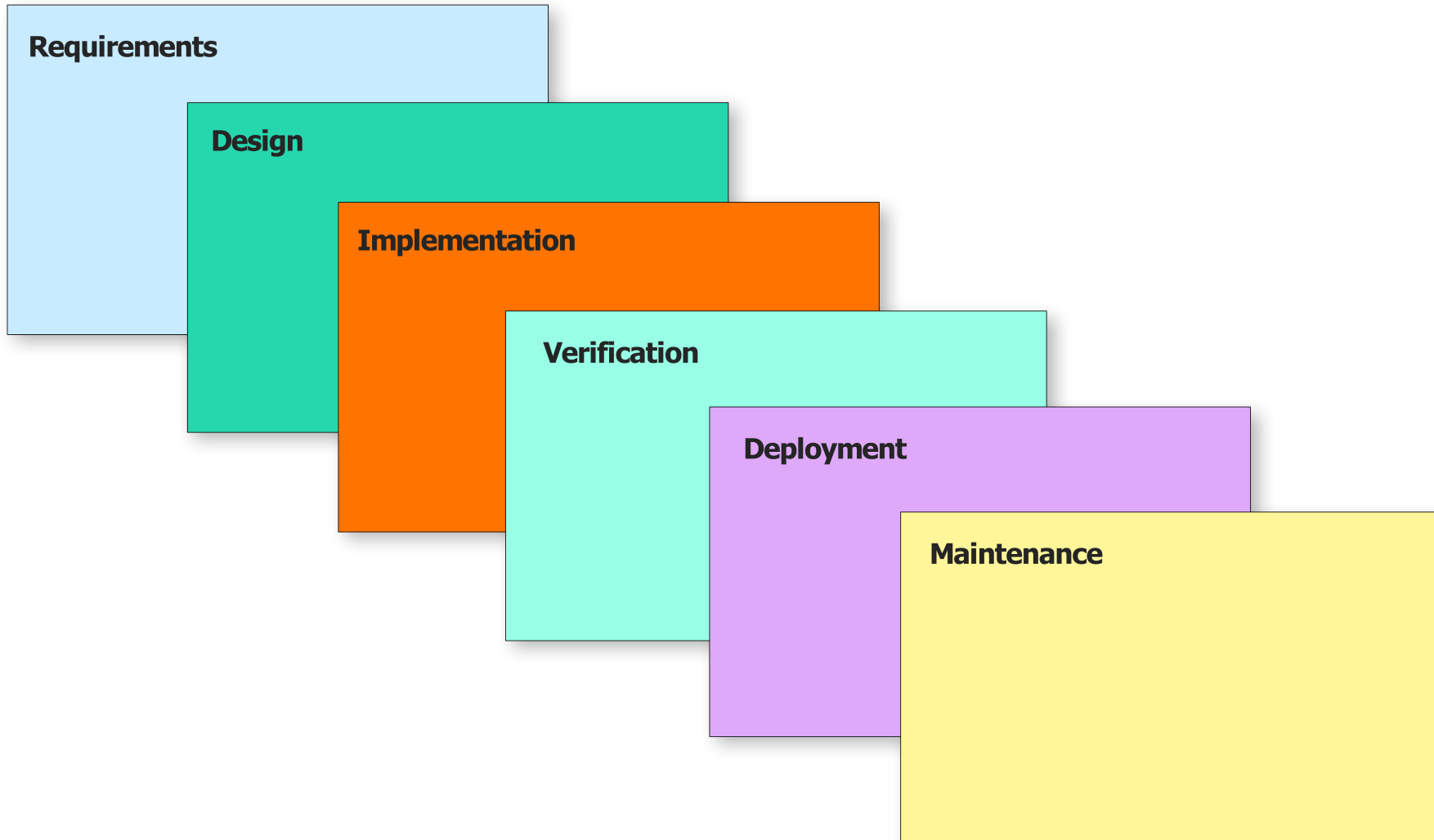
Predictive model: Waterfall



Predictive model: Waterfall with feedback



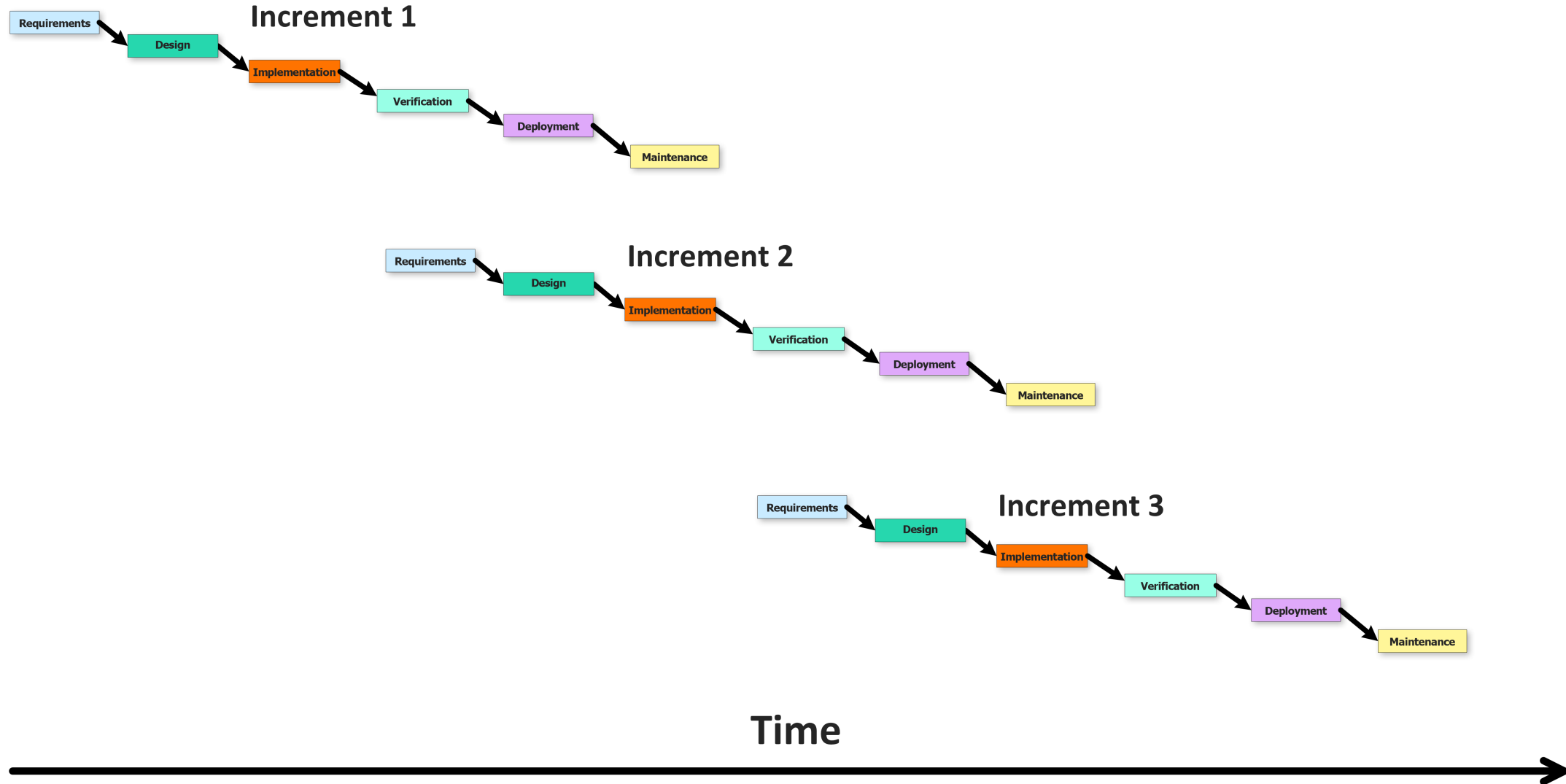
Predictive model: Sashimi waterfall



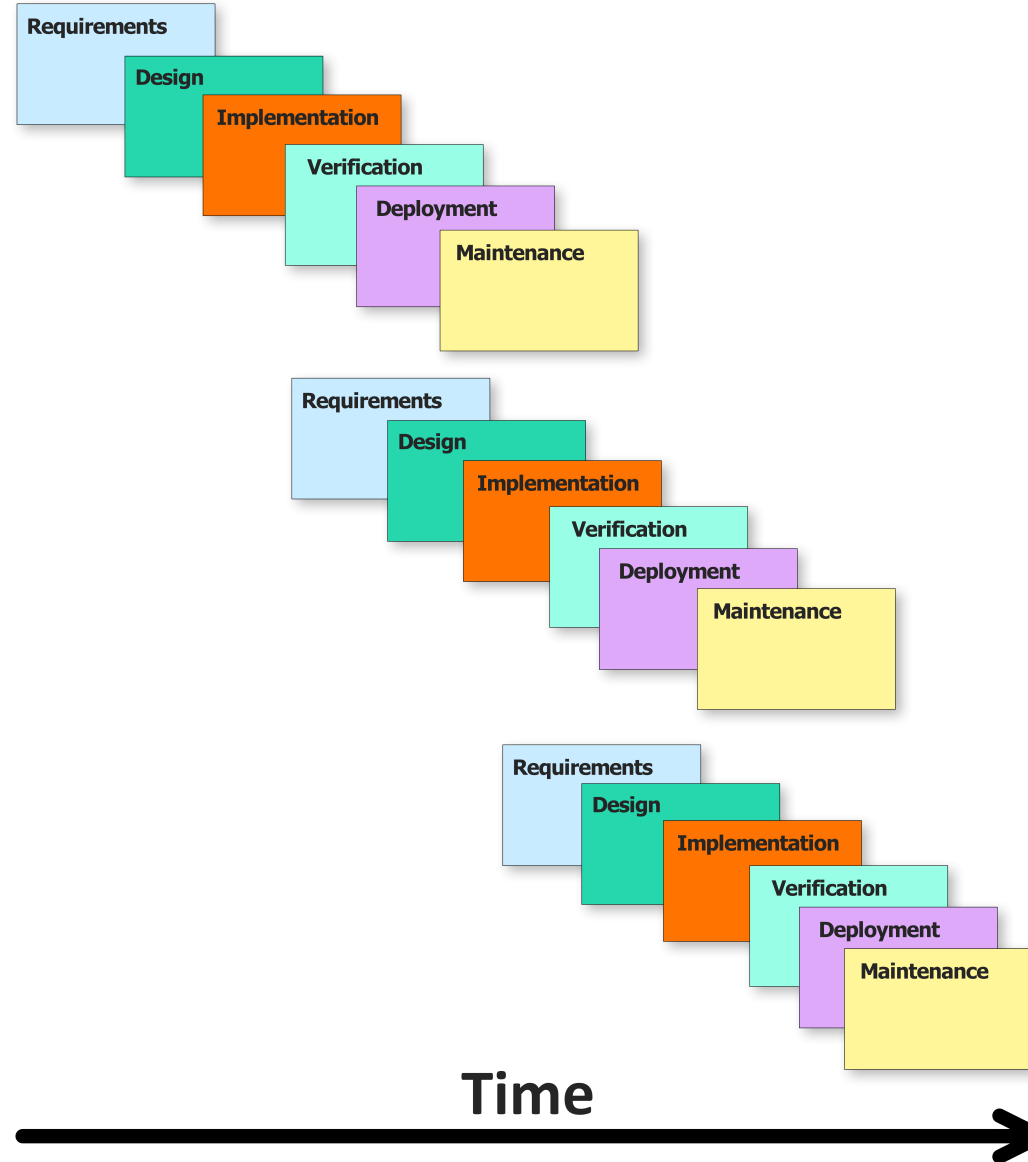
Increments in Predictive models

- A refinement in the predictive model is to develop an application over a series of ***iterations***.
- Iterations provide a feedback-loop in the development process. When all stages in a predictive model are completed, an iteration is completed and a product is produced. The product can then be tested and evaluated by developers and the customers. A new iteration in a predictive model can then begin with feedback from the previous iteration.

Predictive model: Incremental waterfall

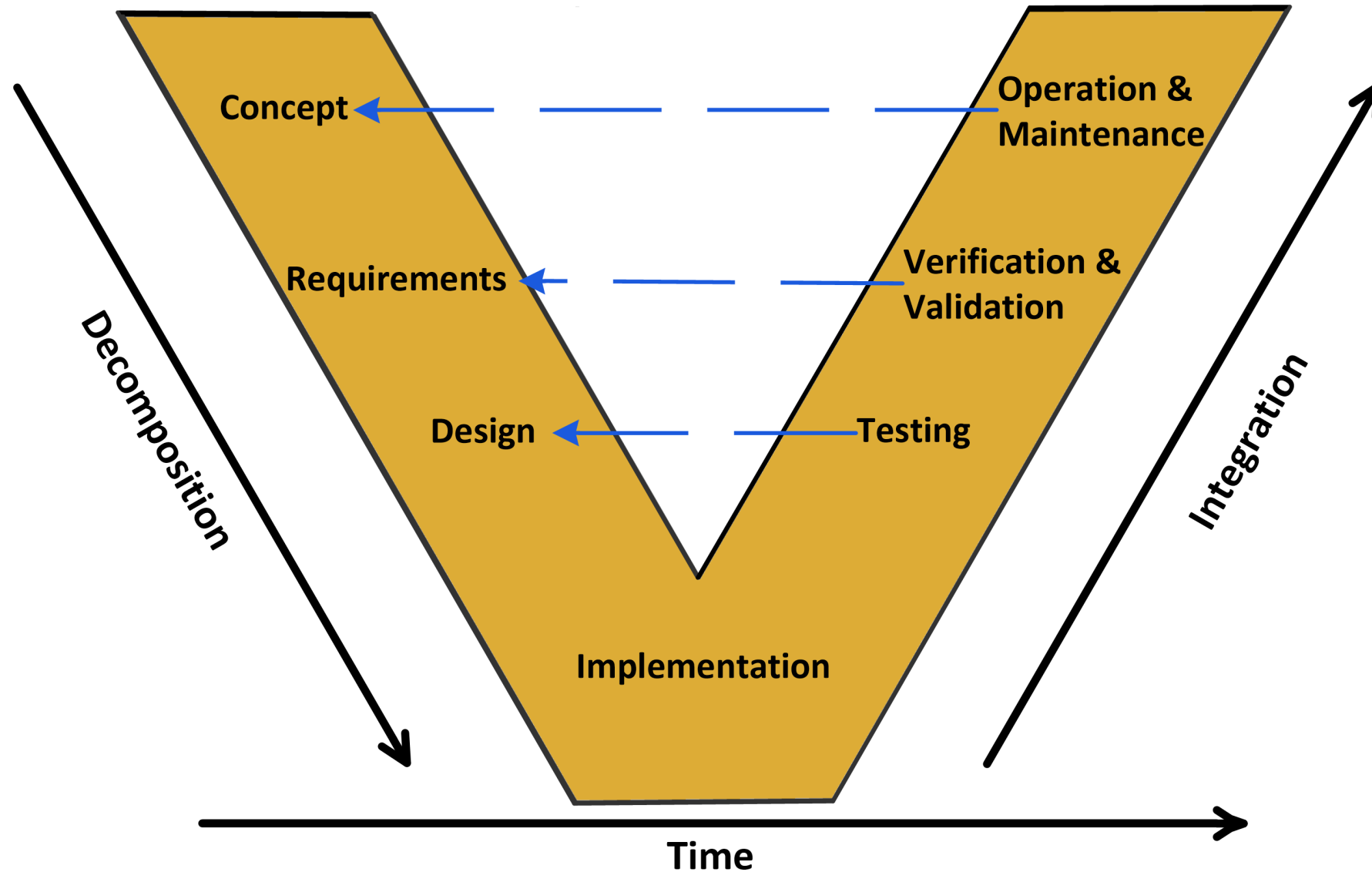


Predictive model: Incremental Sashimi waterfalls



Source: page 437, *Beginning Software Engineering* (2nd edition) by Rod Stephens

Predictive model: V-model



Composition of the V-model

- The left side of the V-model outlines the tasks required to build the project through a process called ***decomposition***.
- The right side of the V-model outlines various stages of completion for the final project through a process called ***integration***.

Software Development Life Cycle (SDLC)

